JOURNAL OF APPLIED
COMPUTER SCIENCE
Vol. 20 No. 2 (2012), pp. 107-117

Low-Complexity Approximation of 8-point Discrete Cosine Transform for Image Compression

Dariusz Puchala[1], **Kamil Stokfiszewski**[1]

[1]*Lodz University of Technology*
Institute of Information Technology
Wólczańska 215, 90-924 Łódź
dariusz.puchala@p.lodz.pl,
kamil.stokfiszewski@p.lodz.pl

Abstract. *In this paper the authors propose a new low-complexity approximation of 8-point discrete cosine transform (DCT) that requires 18 additions and two bit-shift operations. It is shown that the proposed transform outperforms significantly the known transform of the same computational complexity when applied to a JPEG compression stream in practical cases of encoding and decoding of still images. As such, the proposed transform can be effectively used in any practical applications where significant limitations exist regarding the computational capabilities coding and/or decoding devices, i.e. mobile devices or industrial imaging devices.*
Keywords: *discrete cosine transform, fast algorithms, image compression, JPEG standard.*

1. Introduction

Many of the most popular standards of lossy compression of still images use scalar quantization in the domain of the chosen orthogonal transforms. A widely used and an attractive tool for reducing the mean-square error of image degradation in the aforementioned lossy compression scheme is the discrete cosine transform (DCT), for which there exist computationally efficient algorithms and which also allows to obtain a mean-square error performance very close to this, which is optimal for natural images [1]. Since the possibilities of further reduction of the computational complexity of the floating-point DCT are strongly limited, and recent applications such as the Internet and mobile multimedia communications or digital cameras' imaging require highly effective implementations of discrete transforms for image compression, one can observe a non-decreasing interest in finding even more computationally efficient approximate versions of DCT [2, 3, 4, 5]. In particular, one can find many recently proposed low-complexity versions of 8-point DCT such as the one introduced in [2], which is an approximate floating-point DCT formed by proper introduction of zeros into the matrix of signed DCT (SDCT) reported in [3]. It requires 18 additions and two bit-shift operations while SDCT is calculated with 24 additions. In [4] two approximations of DCT are proposed where the coarsest of which requires two multiplications and 18 additions. Lastly, the proposed in [5] matrix $\hat{\mathbf{D}}_1$ describes a low-complexity approximation of DCT with 24 additions and two bit-shift operations. To conclude there is a permanent interest among image compression community in finding ever more efficient implementations of the DCT and it's approximations. Finally, it's worth mentioning that for 8-point 1D DCT with quantization the fastest known algorithm [6] requires 5 multiplications and the theoretical bound for 8-point 1D DCT (without quantization step) is proved to require at least 11 multiplications [7, 8]. Such practical algorithm, with 11 multiplications and 29 additions, has been developed in [9].

2. The proposed 8-point transform

The proposed approximation of 8-point DCT is based on an approach presented in [4]. The main idea presented in [4] was to approximate 8-point DCT with 4-point DCTs applied to four simple Haar transforms calculated as the sum and difference of elements from adjacent pairs of elements in input vector. The coarsest approximation in [4] applied 4-point DCT to the sums of elements, while the differences formed the high-frequency coefficients. In this paper, we also approximate the 4-point DCT that is applied to low-frequency Haar coefficients. As a result, we obtain multiplication-free approximation of 8-point DCT that requires 18-additions and two bit-shift operations. It has the same number of additions as in [4] but multiplications are replaced by bit-shift operations. The proposed transform can be described by the 8x8 matrix **V** of the following form

$$\mathbf{V} = \mathbf{D}\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1.5 & 1.5 & 0.5 & 0.5 & -0.5 & -0.5 & -1.5 & -1.5 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 0.5 & 0.5 & -1.5 & -1.5 & 1.5 & 1.5 & -0.5 & -0.5 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{pmatrix} = \mathbf{D}\,\mathbf{U},$$

where $\mathbf{D} = 1/\sqrt{2}\, diag\left(1/2, 1/\sqrt{5}, 1/2, 1/\sqrt{5}, 1, 1, 1, 1\right)$. It should be noted that the proposed matrix **V** is orthogonal. Hence, the inverse transform equals $\mathbf{V}^{-1} = \mathbf{V}^T = \mathbf{U}^T\mathbf{D}$, where $(\cdot)^T$ denotes matrix transposition. In standard JPEG encoder [10] input images are divided into 8x8 blocks, all blocks are transformed with 2D DCT and the resulting 8x8 matrices in the transform domain are quantized. At the decoder that process is inverted. First, all blocks are dequantized and then transformed with inverse 2D DCT. This scheme is depicted in Fig. 1.

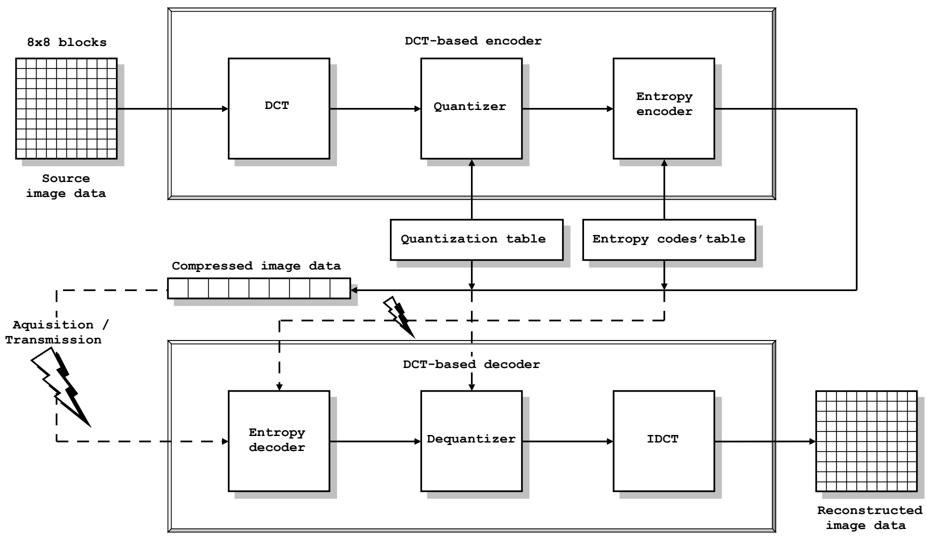

Figure 1. JPEG scheme for compression and decoding of still images

Since 2D DCT is separable [1], it can be calculated with use of 1D DCTs applied to rows and columns of the blocks. Such scheme is depicted in Fig. 1 and is also used during evaluation of the experimental results. Let $\mathbf{X}$ be an 8x8 block matrix of an image and $\mathbf{Y}$ be the block matrix obtained in the transform domain. Then, the forward and inverse transforms can be calculated with use of the proposed low-complexity approximation of DCT as $\mathbf{Y} = \mathbf{V}\,\mathbf{X}\,\mathbf{V}^T = \mathbf{D}\,(\,\mathbf{U}\,\mathbf{X}\,\mathbf{U}^T)\,\mathbf{D}$ and $\mathbf{X} = \mathbf{V}^T\mathbf{Y}\,\mathbf{V} = \mathbf{U}^T(\,\mathbf{D}\,\mathbf{Y}\,\mathbf{D}\,)\,\mathbf{U}$ respectively. Since quantization/dequantization is applied to the block matrices in transform domain, the multiplications required by diagonal matrix $\mathbf{D}$ can be merged into the quantization/dequantization matrices. It means that arithmetic operations required by matrix $\mathbf{U}$ are the only additional operations in the compression process. The $\mathbf{U}$ matrix can be decomposed into a product of four sparse matrices $\mathbf{U} = \mathbf{U}_4\,\mathbf{U}_3\,\mathbf{U}_2\,\mathbf{U}_1$, given in (1) as

$$\mathbf{U}_1 = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{pmatrix}, \quad \mathbf{U}_2 = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix},$$
$$\mathbf{U}_3 = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \quad \mathbf{U}_4 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -0.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \tag{1}$$

On that basis, we can construct the data flow-graph for effective calculation of $\mathbf{U}$ transform presented in Fig. 2. It is easy to verify on the basis of the data flow-graph from Fig. 2, that the computation of the proposed DCT approximation requires 18 additions and two bit-shift operations per single input vector.

3. Experimental results

In order to check the effectiveness of the proposed 8x8 transform, the experiments were performed involving test images and JPEG compression stream

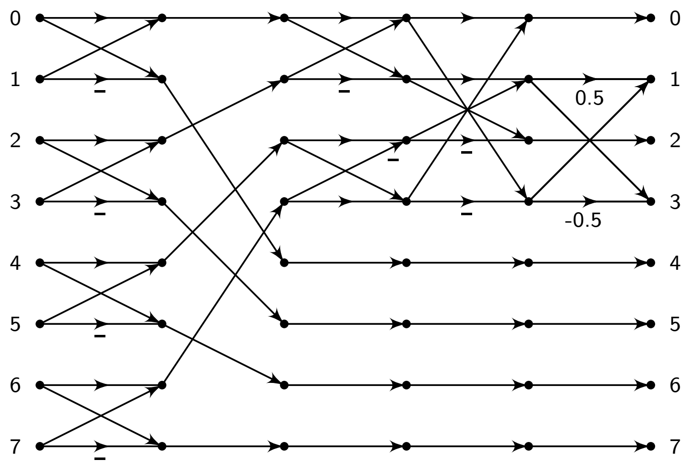

Figure 2. The data flow-graph for the calculation of the **U** transform

(see Fig. 1). Not only the proposed transform was considered but also the approximations of the DCT transform reported in [2, 3, 4, 5]. The results were evaluated for grayscale 512x512 pixel 'Lena' image, shown below in Fig. 3a), and 256x256 pixel 'Cameraman' image, which is depicted in Fig. 3b).

a) b)

Figure 3. Experimental images: a) Lena, b) Cameraman

The results obtained for 'Lena' 512x512 and 'Cameraman' 256x256 images were evaluated in the following variants of the conducted experiments, namely: (i) the encoder and the decoder both use forward and inverse transforms of the same type, (ii) the encoder uses low-complexity transform but the decoder uses inverse DCT (e.g. image is encoded on mobile device but is decoded on personal computer). The results for the third possible variant, i.e. the encoder uses DCT but the decoder uses inverse low-complexity transform (e.g. image from Internet is decoded on mobile device) were analogous to those of the second variant and, hence, are not presented in the paper. In all the experiments the standard quantization and Huffman code tables (see Fig. 1), recommended by the specification of JPEG compression stream, [10] were used.

In the first variant of the conducted experiments the proposed transform obtained very close results to those reported in [4] for an approximate DCT (which requires two multiplications) and performed significantly better for low-bit rates than the approximate transform of the same computational complexity, which was proposed in [2]. The best results were obtained with $\hat{\mathbf{D}}_1$ approximate DCT introduced in [5] which, however, requires 25% more computations than the proposed transform. The worst performance was reported in the case of SDCT [3], which is also more computationally complex than the proposed transform. The results of the experiment (i), obtained for 'Lena' image, are presented in Fig. 4, where standard PSNR measure of image degradation resulting form lossy compression process is given by the following equations

$$\mathrm{PSNR} = 10 \cdot \log_{10} \frac{255^{2}}{\mathrm{MSE}} ,$$

$$\mathrm{MSE} = \frac{1}{N^{2}} \sum_{i=1}^{N} \sum_{j=1}^{N} \left(x_{ij} - \hat{x}_{ij} \right)^{2} ,$$

where N denotes horizontal/vertical experimental images' dimensions (Fig. 3a,b) and x_{ij} and $\hat{x}_{ij}$ denote the original and the reconstructed image's pixels respectively. The bpp rate is a measure of how many bits per pixel are needed on average to store the file containing all of the compressed image data.

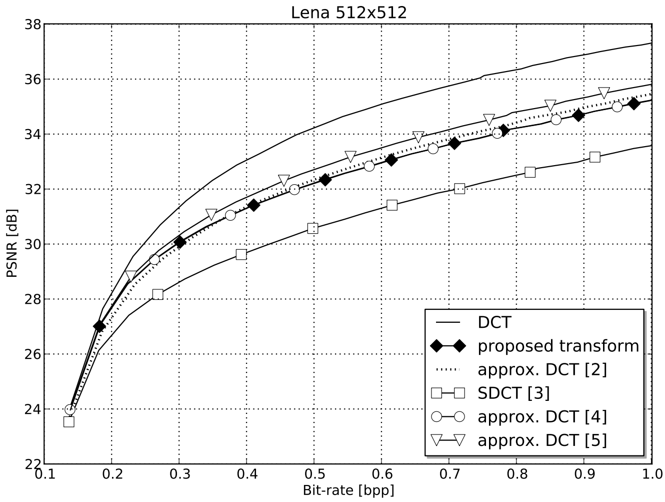

Figure 4. PSNR of the transforms considered in experiment (i) for Lena image

In the second (and the third) variant of experiments the proposed transform outperformed significantly the low-complexity transform from [2]. The best results were obtained with $\hat{\mathbf{D}}_1$ from [5] and the performance of SDCT depended highly on the contents of images. Lastly, the approximate DCT from paper [4] allowed to obtain results very close to those of the proposed transform.

Table 1. Results of the experiment (ii) for Lena image

PSNR / bpp	**0.4**	**0.6**	**0.8**	**1.0**	**1.2**	**1.4**
proposed transform	31.550	31.913	31.786	31.615	31.457	31.322
approx. DCT [2]	27.820	27.713	27.589	27.480	27.391	27.314
approx. DCT [4]	31.586	31.954	31.866	31.694	31.544	31.401

Table 2. Results of the experiment (ii) for Cameraman image

PSNR / bpp	**0.4**	**0.6**	**0.8**	**1.0**	**1.2**	**1.4**
proposed transform	24.930	24.929	24.734	24.526	24.379	24.257
approx. DCT [2]	22.073	22.030	21.915	21.811	21.745	21.677
approx. DCT [4]	24.923	24.958	24.749	24.563	24.409	24.282

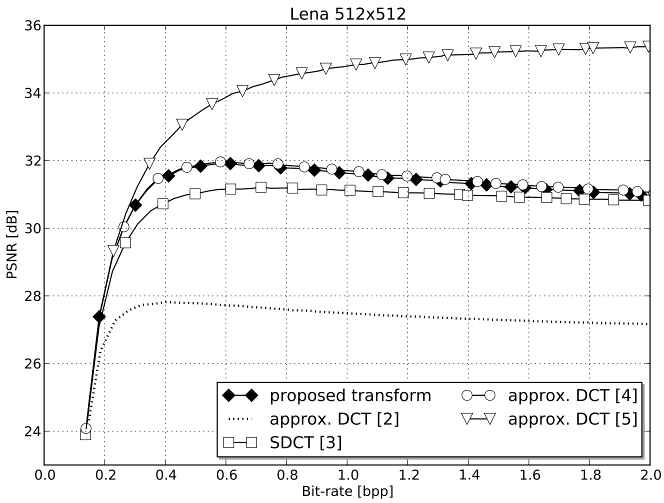

Figure 5. PSNR of transforms considered in experiment (ii) for Lena image

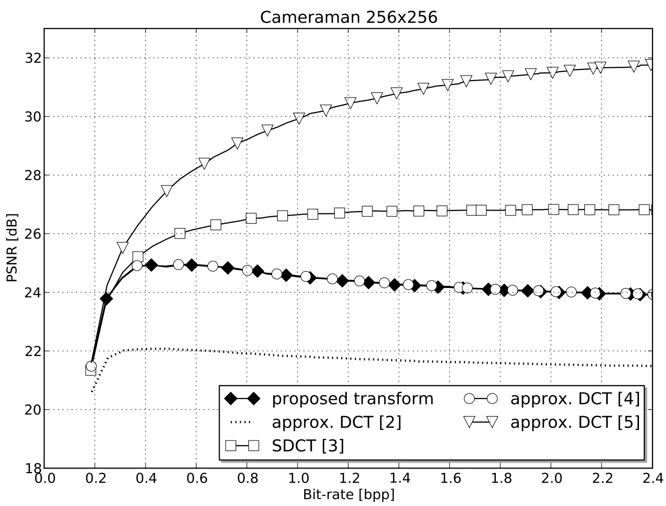

Figure 6. PSNR of transforms considered in experiment (ii) for Cameraman image

Fig. 5 and Fig. 6 show graphs of the PSNR versus bit rate (bpp) obtained for all of the considered transforms, which took part in the experiment (ii), while Tab. 1 and

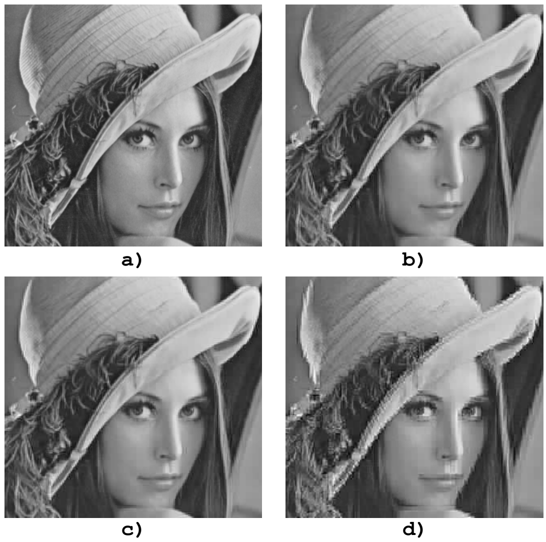

Figure 7. Test fragments of Lena image for transforms in experiment (i)

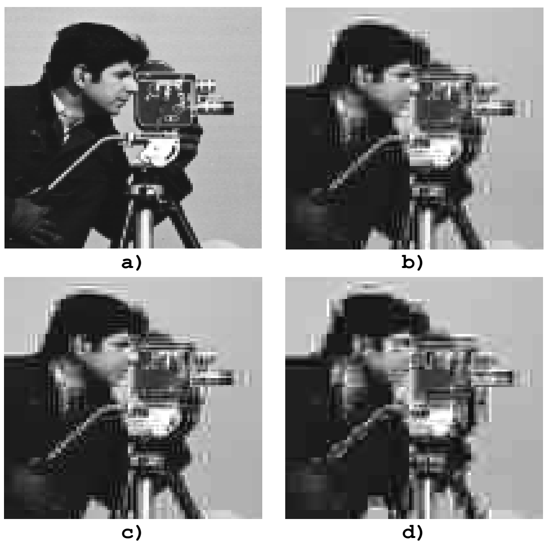

Figure 8. Test fragments of Cameraman image for transforms in experiment (ii)

Tab. 2 show results obtained for the three most interesting of the considered transform approximations, namely, the proposed one, the transform introduced in [2], which has the same computational complexity as the transform proposed in this article and finally the transform derived in [4], which is the closest, in the sense of the PSNR measure, to the proposed transform, and which on the other hand is less computationally efficient. In Fig. 7 and Fig. 8 selected fragments of the 'Lena' and 'Cameraman' images, that took part in the experiment (ii), are shown, wherein, images a) represent the original image fragment, images b) represent the same fragment compressed with the use of the proposed transform and decoded with DCT, images c) are fragments coded using the approximation [4] and decompressed with DCT and images d) show fragments that where compressed with the transform presented in [2] and decoded also with the help the DCT. Looking at the above results, their both numerical and visual evaluations lead to the conclusion that the proposed transform has significantly better qualitative performance than the transform proposed in [2], which is of the same computational complexity and, on the other hand, the proposed transform's qualitative performance is almost equal to the transform derived in [4], which is more computationally demanding.

4. Conclusions

A low-complexity approximation of 8-point discrete cosine transform (DCT) for image compression is proposed. It is obtained by appropriate modification of approximate DCT from [4]. As a result the multiplication-free transform that requires 18 additions and two bit-shifts is obtained. It has been shown experimentally with test images and JPEG compression stream that the proposed transform allows to obtain the same results as more computationally complex transform [4]. Moreover, it outperforms the well-known transform of the same complexity [2] for both low-bit rates compression and asymmetric compression where encoding or decoding transform is substituted by DCT or its inverse respectively.

References

[1] Rao, D. R. and Yip, P., *Discrete Cosine Transform*, San Diego, CA: Academic, 1990.

[2] Bouguezel, S., Ahmad, M. O., and Swamy, M. N. S., *Low-complexity 8x8 transform for image compression*, Electronics Letters, Vol. 44, No. 21, 2008.

[3] Haweel, T. I., *A new square wave transform based on the DCT*, Signal Processing, Vol. 81, No. 11, 2001, pp. 2309–2319.

[4] de Queiroz, R. L., *DCT approximation for low bit rate coding using a conditional transform*, In: IEEE International Conference on Image Processing, Vol. 1, 2002, pp. 237–240.

[5] Lengwehasatit, K. and Ortega, A., *Scalable variable complexity approximate forward DCT*, IEEE Trans. Circuits Syst. Video Technol., Vol. 14, No. 11, 2004, pp. 1236–1248.

[6] Arai, Y., Agui, T., and Nakajima, M., *A Fast DCT-SQ Scheme for Images*, IEICE Transactions, Vol. 71, No. 11, 1988, pp. 1095–1097.

[7] Duhamel, P. and H'Mida, H., *New 2^n DCT Algorithms suitable for VLSI Implementation*, In: IEEE International Conference on Acoustics, Speech and Signal Processing, April, 1987, pp. 1805–1808.

[8] Heidemann, M. T. and Burrus, C. S., *Multiply/Add Tradeoffs in Length 2^n FFT Algorithms*, In: IEEE International Conference on Acoustics, Speech, and Signal Processing, March, 1985, pp. 780–783.

[9] Loeffler, C., Ligtenberg, A., and Moyschytz, G. S., *Practical Fast 1-D DCT Algorithms with 11 Multiplications*, In: IEEE International Conference on Acoustics, Speech, and Signal Processing, May, Vol. 2, 1989, pp. 988–991.

[10] *Digital compression and coding of continuous-tone still images*, Tech. Rep. T.81, International Telecommunication Union, 1992.